

OFFICES  
McGuireWoods LLP  
1750 TYSONS BOULEVARD, SUITE 1800  
MCLEAN, VIRGINIA 22102

APPLICATION  
FOR  
UNITED STATES  
LETTERS PATENT

Applicants: **Paul Snyder**  
For: **DYNAMIC TRANSACTION  
CONTROL WITHIN A HOST  
TRANSACTION PROCESSING SYSTEM**  
Docket No.: **00240295US**

# **DYNAMIC TRANSACTION CONTROL WITHIN A HOST TRANSACTION PROCESSING SYSTEM**

## **DESCRIPTION**

### **BACKGROUND OF THE INVENTION**

#### *Field of the Invention*

**[0001]** This invention relates to transaction processing systems, and more particularly, to autonomic control and administration of individual transactions or groups of transactions based upon their unique current resource usage characteristics relative to the present status of one or more present characteristics of the transaction processing system or the host computer system.

#### *Background Description*

**[0002]** Transaction processing systems serve as a basis for electronic commerce. In electronic commerce two or more entities electronically process specific tasks related to commerce ranging from purchase and payment to banking transactions. Examples of electronic commerce include purchase and payment transactions using credit and debit cards, paying bills online and the handling of return merchandise credits. Transaction processing systems also facilitate the accessing of data across a network, such as the

Internet. Browsing merchandise on a vendor's website, obtaining stock quotes from a financial service institution's website and checking sporting event scores at a news website are examples of such accesses. Other examples of routine network transaction events include the interchange of data that occurs during online gaming, downloading product updates from a software vendor's server and the exchange of email.

**[0003]** Prior to the days of the Internet, an example of a transaction processing system would be a given corporation's internal users accessing a host mainframe processor, with individual transactions being serviced via an executing software "transaction monitor", such as, for example, IBM's CICS (Customer Information Control System). With the advent of the Internet, an example of a transaction processing system includes the server or servers that host a given Internet website, along with the underlying hardware and software infrastructure, frequently including an "application server" such as IBM's WebSphere Application Server.

**[0004]** Transactions that are serviced by a transaction processing system may come from any number of sources, examples of which include users accessing the system from a home or office personal computer, personal digital assistant (PDA), network-enabled cellular devices and automated teller machines (ATM's). Additionally, transaction processing systems may also be accessed by other systems, such as partner transaction processing systems, interactive voice response systems, or any other automated entity that has access to the transaction processing system through a network.

[0005] For a variety of reasons such as application errors, hardware faults and unintended use of the transaction processing system, there is a chance that a transaction or group of transactions may take on characteristics that are outside of the design specifications of the transaction processing system. Actions commonly taken to eliminate the undesirable workload from the system could range from purging the transaction to shutting down and restarting the entire transaction processing environment.

[0006] In other cases it may simply become desirable to favor one type of work versus another type based upon current general system conditions, such as high utilization or the like. Current practice facilitates the remediation of these conditions through binary evaluation. For example, if a given transaction on a transaction processing system has consumed more than a predetermined number of CPU seconds, then the offending transaction is terminated. A termination of an offending transaction may also occur, for example, if a given transaction were consuming more than some predetermined amount of electronic storage.

[0007] Current practice also allows for indiscriminate termination of transactions in the event of an alert on the transaction processing system. An example of the type of condition that would trigger such an alert is when a short on storage event occurs within the transaction processing system.

**[0008]** A shortcoming of the current practices arise from the indiscriminate nature of transaction administration. In practice, transaction processing systems purge transactions that exceed some arbitrary limit, without regard to whether the limit that was exceeded is presently constrained on the transaction processing system.

**[0009]** A further shortcoming is when remedial action is taken or initiated to address an alert from the transaction processing system, but the nature of the action is poorly targeted and/or overly drastic, affecting transactions that have little or no bearing on the alert, itself. Currently in transaction control facilities, autonomic corrective action is typically ill-targeted, poorly timed, and affects the user community too broadly.

## **SUMMARY OF THE INVENTION**

**[0010]** In an aspect of the invention, a method is provided for managing a transaction processing system. The method comprises defining at least one criterion which is at least a workload characteristic and defining at least one threshold metric for each of the at least one criterion. The method further comprises defining at least one trigger action in response to the at least one threshold metric and performing the at least one trigger action in response to the at least one threshold metric being met.

[0011] In another aspect of the invention, a method is provided for managing a system.

The system comprises the steps of determining current conditions of a workload characteristic, evaluating the current conditions of the workload characteristic and dynamically adjusting system administration criteria based on a threshold metric associated with the current conditions of the workload characteristic.

[0012] In another aspect of the invention, a system is provided for managing a transaction processing system. The system comprises a means for defining at least one criterion, wherein the at least one criterion is a workload characteristic of the transaction processing system and a means for defining at least one threshold metric for each of the at least one criterion. The system further comprises a means for defining at least one trigger action in response to the at least one threshold metric.

[0013] In another aspect of the invention, a system is provided for managing a transaction processing system. The system comprises a means for determining current conditions of at least a workload characteristic, a means for evaluating the current conditions of at least the workload characteristic and a means for dynamically adjusting system administration criteria based on a threshold metric associated with the current conditions of at least the workload characteristic.

[0014] In another aspect of the invention, a computer program product is provided comprising a computer usable medium having readable program code embodied

in the medium. The computer program product includes a first computer code to define at least one criterion, wherein the at least one criterion is a workload characteristic of the transaction processing system and a second computer code to define at least one threshold metric for each of the at least one criterion. Further included are a third computer code to define at least one trigger action in response to the at least one threshold metric and a fourth computer code to perform the at least one trigger action in response to the at least one threshold metric being met.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0015] Figure 1 is a block diagram illustrating a typical distributed data processing system in which the present invention may be implemented;

Figures 2A-2E are flow diagrams of an embodiment showing steps of using the invention; and

Figure 3 is a flow diagram showing an embodiment of using an aspect of the invention.

### **DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION**

[0016] In the system and method of the invention, an autonomic extension of a transaction processing system dynamically adjusts system administration criteria based

upon the current conditions of the transaction processing system. The invention provides for the ability of system designers and administrators to apply greater autonomic logic to the governance of a transaction processing system. The invention also allows for very granular autonomic administrative action relative to varying conditions within the transaction processing system as well as a host of other advantages as discussed herein. System designers and administrators are enabled to take into consideration fluctuating conditions throughout the server/system when developing autonomic control models.

**[0017]** Figure 1 is a block diagram illustrating a typical distributed data processing system in which the invention may be implemented. The distributed data processing system 100 comprises a number of computers, connected by a network 120. In one implementation, server computer 130 is connected to the network 120 along with a storage unit 150 and client computers 160, 170 and 180. In the depicted example, distributed data processing system 100 may be the Internet, with the network 120 representing a world-wide collection of networks and gateways that use the transmission control protocol over internet protocol (TCP/IP) suite of protocols to communicate with one another. An embodiment of this invention may involve server 130 to server 140 interactions, or storage to storage interactions.

**[0018]** The invention may be divided into four recurring stages: (1) obtain system level metrics, (2) obtain transaction level metrics, (3) perform evaluations, and (4) perform action based on the evaluations. Restated, as a processing cycle begins, the

invention gathers salient details regarding the state of the transaction processing system and related environment, i.e., the system level metrics. The second stage of processing involves the collection of details or metrics regarding individual executing transactions, i.e., transactional level metrics. The third stage of processing performs evaluations which are defined in the interval criterion tables, which may result in stage four, which includes performing an administrative action. Embodiments of the invention are referenced herein as “facility” or “the facility” and may be a software extension of a transaction processing system.

[0019] In embodiments, the components to support the four stages may include: (i) a component to collect system and transaction information that is germane to an administrator’s decision making process, (ii) a component to perform complex evaluations between the collected data and the invention’s configuration data, and (iii) a component to execute an action that is germane to an administrator’s functions. (See discussion, for example, of Figures 2A-2E).

[0020] The invention also includes implementation of an interval criterion matrix and it typically may be a source of configurable data used by the invention and may be created by an administrator or accessed from a pre-built electronic source. An example of the interval criterion matrix is shown in Table 1 and is described in further detail below with reference to Figures 2A-2E.

TABLE 1  
Example Interval Criterion Matrix

	(601) System Level Metric	(602) Transaction Identifier	(603) Transaction Level Metric	(604) Facility Action
1	Average System Processor Utilization is in the 50%-75% range	AA*	Transaction Processor Utilization is greater than 30 Seconds	Reduce Priority of Transaction
2	Average System Processor Utilization is in the 50%-75% range	AA*	Transaction Processor Utilization is greater than 45 Seconds	Reduce Priority of Transaction & Quiesce Transaction for 10 Seconds
3	Average System Processor Utilization is in the 50%-75% range	AA*	Transaction Processor Utilization is greater than 60 seconds	Terminate Transaction
4	Average System Processor Utilization is in the 75%-99% range	AA*	Transaction Processor Utilization is greater than 10 Seconds	Reduce Priority of Transaction
5	Average System	AA*	Transaction Processor	Reduce Priority of

	Processor Utilization is in the 75%-99% range		Utilization is greater than 15 Seconds	Transaction & Quiesce Transaction for 10 Seconds
6	Average System Processor Utilization is in the 75%-99% range	AA*	Transaction Processor Utilization is greater than 20 Seconds	Terminate Transaction
7	Average System Processor Utilization is 100%	AA*	Transaction Processor Utilization is greater than 2 Seconds	Reduce Priority of Transaction
8	Average System Processor Utilization is 100%	AA*	Transaction Processor Utilization is greater than 4 Seconds	Reduce Priority of Transaction & Quiesce Transaction for 10 Seconds
	Average System Processor Utilization is 100%	AA*	Transaction Processor Utilization is greater than 6 Seconds	Terminate Transaction

[0021] The configuration data shown in Table 1 comprises a matrix that includes a “System Level Metric” 601. The system level metric may be a single or progressive variable (e.g., ranges) relative to a measurement of an aspect of the transaction processing

system. The matrix of Table 1 further includes a "Transaction Identifier" 602 which may, for example, provide selection criteria for a single transaction or plurality of transactions. Table 1 further includes a "Transaction Level Metric" 603 and a "Facility Action" 604. The Transaction Level Metric 603 may be a single or progressive variable relative to an aspect of the included transaction, and the Facility Action 604 may be, for example, a reference to an action to occur should the variable evaluation result be positive. In embodiments, system-level criterion (e.g., 601 and 604) may be in a separate table.

**[0022]** By way of example, using the data of Table 1, when the System Level Metric has an average system processor utilization in the range of 50%-75%, and the Transaction Level Metric 603 is greater than 30 seconds, the Facility Action 604 may reduce the priority of a current transaction. Of course, it should be understood that other scenarios may also be implemented by the invention as discussed herein and as now should be understood by those skilled in the art. Now as it should be understood, within the overall process of the invention, various system-level metrics of the transaction processing system and the supporting hardware, software and networking environment are sampled. Examples of such system-level metrics may include but not limited to, processor utilization, memory utilization, storage utilization, load upon the input/output subsystem(s) and load upon the network interfaces. In embodiments, a list of salient metrics to be collected and a reference to a logical process that performs collection of each of these metrics, exists on an interval criterion data source (e.g., a database).

**[0023]** Also within the overall process a method is provided for gathering details of each transaction executing on the transaction processing system at a point in time. As with the system-level metrics, this method accesses a list of salient metrics to be collected and a reference to a logical process that performs collection of each of these metrics from the interval criterion data source. The invention processes through each individual transaction in some order, which may be a sequential order.

**[0024]** As transaction-level metrics become available, thresholds that may be stored in the interval criterion matrix are evaluated. A feature of the invention includes the incremental nature of both the system-level and transaction-level metrics. For example, system-level processor utilization levels of 30%, 40%, 50% and 60% may all have differing thresholds on the transaction-level side of the metric's evaluation. Furthering the example, transaction-level processor utilization of 10 CPU seconds may be considered excessive at 50% system-level processor utilization, transaction-level processor utilization of 8 CPU seconds may be considered excessive at 60% system-level processor utilization.

**[0025]** When both a system-level and transaction-level evaluation result in a "true" result for a given interval criterion entry, the invention invokes the action specified in the same interval criterion entry. Again, as with other logic points indicated by the interval criterion matrix, in embodiments, the action is stored in the interval criterion data source as a reference to a logical process that performs said action.

[0026] Figures 2A-3 are flow diagrams of an embodiment showing steps of using the invention. Figures 2A-3 may equally represent a high-level block diagram of components of the invention implementing the steps thereof. The steps of Figures 2A-3 may be implemented on computer program code in combination with the appropriate hardware. This computer program code may be stored on storage media such as a diskette, hard disk, CD-ROM, DVD-ROM or tape, as well as a memory storage device or collection of memory storage devices such as read-only memory (ROM) or random access memory (RAM). Additionally, the computer program code can be transferred to a workstation over the Internet or some other type of network. Figures 2A-3 may be implemented, for example, using the components of Figure 1.

[0027] Referring to Figures 2A-2E, upon activation of the facility, at step 200, the runtime parameters 210 of the facility (which may be maintained in a file or database, or may be hard coded or hardwired into the invention itself) are loaded at step 220 and validated at step 230. Depending upon the results of validation step 230, a check is made at step 240 to determine whether processing should continue, based upon the qualification criteria present in the embodiment of the invention. Should validation fail, the facility terminates at step 270. After step 250 successfully completes initialization of the facility, the system enters a nominal mode of processing at step 260, which may be regulated in intervals.

**[0028]** Continuing with Figure 2B, upon the start of a nominal processing cycle at step 310, the frequency may be governed by an interval controller 380, discussed below. At step 320, the facility collects data regarding the present or current workload characteristics, status, or “health”, of the transaction processing system. The facility accesses a pre-provided list of characteristics 330 for which data is to be collected. Examples of the source of this data may include the facility’s runtime parameters 210 or the storage mechanism used to maintain the interval criterion (e.g., 420) which may be a database, for example. For each characteristic of the transaction processing system or hosting server for which data may be collected, there may be a method to collect such data, such as dedicated routines and/or registry for collecting the data. Examples of where this logic resides include integrated routines 340, typically within the facility itself, as distinct separate executable logic 350 or definitions interpretable by the facility 360. This process continues until, at step 370, it is determined that there are no further collection points for which to collect data.

**[0029]** Once a list of all relevant system metrics is collected and made available, the facility is prepared to begin evaluation of the interval criterion. In embodiments, the invention may perform three levels of dynamic administration:

1. System Level – Dynamic actions based upon system-level health characteristics.
2. Transaction Level – Dynamic actions based upon transaction-specific characteristics.

3. Multi-Transaction Level – Dynamic administrative actions based upon the aggregate characteristics of the current transaction mix executing on a server.

**[0030]** Continuing with Figure 2C, at step 410, system-level criterion is evaluated from the interval criterion data source 420. The system-level evaluation step 410 may be performed directly against the data gathered at step 320. At step 430, a check is made whether the evaluation results in a required action. If not, at step 440, the facility determines if there are more system-level evaluations to be performed. If there are more system-level evaluations, the process continues at step 410 with the next evaluation. Otherwise the facility proceeds to the next series of evaluations. Should the result at step 430 be positive (i.e., requiring action), the action defined by the interval criterion data source 420 is carried out at step 450, using the logic of the interval criterion actions 460, which is associated with the interval criterion that was evaluated at step 410, and may be executed within the software of the facility itself.

**[0031]** In other embodiments the action defined in 460 may be provided through other mechanisms, such as sending messages to other entities to perform an action. Examples of such an action include informing a peer server that the triggering server is now available to accept work, alerting a remote operator of an anomalous condition, or triggering a diagnostic trace on a Storage Area Network unit. After the triggered action is carried out (or instigated) at step 450, the process proceeds to step 440 to determine if there are more system evaluations.

[0032] System-level criterion evaluations deal with characteristics of the system, which may be the transaction processing system of the host server. By way of example, a representation of a system-level criterion tree based on the present/current state of "SYSTEM MEMORY UTILIZATION" would be:

- A- If SYSTEM MEMORY UTILIZATION exceeds 60%, then perform action SLSMU\_1.
- B- If SYSTEM MEMORY UTILIZATION exceeds 80%, then perform action SLSMU\_2.
- C- If SYSTEM MEMORY UTILIZATION exceeds 90%, then perform action SLSMU\_3.
- D- If SYSTEM MEMORY UTILIZATION exceeds 95%, then perform action SLSMU\_4.

For each action identifier (e.g., SLSMU\_1, SLSMU\_2, etc) there is an associated action defined in the interval criteria actions 460. One of ordinary skill in the art should recognize that any number of action identifiers may exist, as necessary.

[0033] Continuing with Figure 2D, transaction-level criterion evaluation may be performed starting at step 500 where the facility acquires a list of currently executing transactions from the transaction processing system, which may in embodiments, be a list of currently executing transactions stored for reference in a Stored Transaction List 590. At step 510, for each transaction on the list, a series of relevant details is collected or obtained. At step 520, transaction level evaluation is performed against the data collected in step 510 using interval criterion matrix 530.

**[0034]** At step 540, a determination is made whether any action is necessary based upon the evaluation at step 520. If no action is required (e.g. negative evaluation), at step 570, the facility determines if there are more transaction-level evaluations to be performed and if so, the process continues with the next evaluation at step 520. Should the result of step 540 require action (e.g. be positive) then, at step 550, action defined by the interval criterion data source 560 is carried out. This action may be carried out within the software of the facility, itself, or may alternatively cause action to be triggered and carried out by another system component. After the triggered action is carried out at step 550, at step 570, the process proceeds to the next transaction-level evaluation check. When there are no further transaction-level evaluations remaining to be performed at step 570, at step 580, the facility determines whether any transactions remain for processing. If so, the next transaction on the list 590 is selected, and processing continues at step 510.

**[0035]** In one implementation and referring to Table 1 and the flow of Figures 2A-2E, transaction-level criterion evaluations deal with the characteristics of individual transactions relative to the present/current characteristics of the overall system, be it the transaction processing system or a host server. An example representation of a transaction-level criterion matrix is given in TABLE 1. The first element of this matrix is System Level Metric 601, which may be obtained in step 320 (Figure 2B). The matrixes second element is Transaction Identifier 602.

**[0036]** The transaction processing system may identify transactions via 4-byte alphanumeric identifiers (or other identification technique). The asterisk in the example Transaction Identifiers 602 in Table 1 indicates a 'wild card', these Transaction Level Criterion affect only transactions that have identifiers starting with "AA". Use of Transaction Identifier 602 in this example illustrates that Transaction Level Criterion do not necessarily need to affect all transactions on a transaction processing system, and may be selectively qualified via any convenient mode of unique identification for a given subset of transactions. The Transaction Level Metric 603 indicates which present characteristic of the transaction to evaluate and the value to evaluate said characteristic against, should the condition listed under System Level Metric 601 be met. For conditions that match criterion System Level Metric 601, Transaction Identifier 602, and Transaction Level Metric 603, the Facility Action 604 is triggered. For clarity within this example, the actions to be invoked under Facility Action 604 are described in plain text. In other embodiments, the Facility Action 604 may be carried on the Transaction Level Criterion Matrix 530 as an identifier referencing a logical process that may be embodied in a form executable by the facility, illustrated as Interval Criterion Actions 560 (Figure 2D).

**[0037]** Multi-transaction criterion evaluations deal with the characteristics of groups of transactions relative to the overall system's present/current characteristics. These evaluations are similar to transaction level criterion evaluation, with the added concept of "transaction groups", which may be built and administered by an

administrator or pre-built and obtained from an electronic source. Examples of such transaction groups are provided in TABLE 2.

TABLE 2  
Example Multi-Transaction-Level Criterion Matrix

	<b>801 System Level Metric</b>	<b>802 Transaction Grouping Method</b>	<b>803 Transaction Group Level Metric</b>	<b>804 Facility Action</b>
1	None	Transactions from the same IP SubNet	Count of transactions in group exceeds 10	Block further transactions from that IP SubNet
2	Current Task Count is in the 70%-80% range of the Maximum Tasks Allowed	Transactions executing search functions	Transaction Group accounts for greater than 50% of current transactions	Terminate transactions within this transaction group
3	Current Task Count is in the 80%-100% range of the Maximum Tasks Allowed	Transactions executing search functions	Transaction Group accounts for greater than 50% of current transactions	Terminate transactions within this transaction group
4	Current Task Count is in the	Transactions routed from the	Transaction Group accounts for greater	Cease accepting transactions from the

	90%-100% range of the Maximum Tasks Allowed	same server	than 25% of current transactions	server & request server to cease forwarding transactions to this server
--	---	-------------	-------------------------------------	---

**[0038]** The concept of transaction groups permits aggregated accounting of transactions over similar or related entities (e.g., same server, search functions, etc.) or common identifier (e.g., IP subnet, transmission medium, etc.) and applying a transaction group level metric to the aggregated entities. This is illustrated by way of example, by referring to Figure 2E, at step 700 the facility acquires a list of aggregate transaction groups from the Interval Criterion tables 790 which defines aggregations of related transaction types or characteristics.

**[0039]** At step 710, for each aggregate transaction group, a series of relevant details is collected, optionally referencing the Stored Transaction List 590 for efficiency reasons. At step 720, transaction-group level evaluation is performed against the data collected at step 710 using interval criterion 730 (e.g., Table 2). At step 740, a determination is made whether action is necessary. Should no action be required (e.g., the result is negative), at step 770, the facility determines if there are more transaction-group level evaluations to be performed. If there are more transaction-group level evaluations, the process continues with the next evaluation at step 720. Should the result

of step 740 determine that action is required (e.g., be positive) then at step 750, the action defined by the interval criterion data source 760 is carried out. The logic of the action, which is associated with the interval criterion that was qualified at step 730 may be carried out within the software of the facility itself. In other embodiments the action logic may trigger action by other components.

[0040] After the triggered action is carried out at step 750, the process proceeds to the next transaction-group level evaluation check at step 770. When, at step 770, there are no further transaction-group level evaluations remaining to be performed, at step 780, the facility determines if any transactions-group evaluations remain for processing. If so, the next transaction-group evaluation on the list 590 is selected, and processing continues at step 710.

[0041] Otherwise, processing proceeds to an Interval Controller 380 that, in this embodiment, halts processing for a set period of time, for example sixty seconds, before allowing processing to continue. Other time intervals may be chosen based on circumstances. In a further aspect, the Interval Controller acts upon variable timers which may change based upon the results of prior scan cycles. In yet another aspect, the Interval Controller may resume processing based upon one or more system characteristics, such as when average processor utilization is greater than 75%, or, when the number of connections to this transaction processing system exceed 2000 (or other

parametric criterion). Alternatively, the Interval Controller may be substantially absent or by-passed, with implies continuous processing of the invention.

**[0042]** Referring again to Table 2, the first element of this matrix is System Level Metric 801, which may be obtained previously at step 320 (Figure 2B). The matrixes second element is Transaction Grouping Method 802. For clarity within this example the Transaction Grouping Method 802 is described in plain text. In other embodiments, the Transaction Grouping Method 802 may be carried on the Transaction Level Criterion Matrix 730 as an identifier referencing a logical process that would be embodied in a form executable by the facility, illustrated as Interval Criterion Actions 760 (Figure 2F). The Transaction Group Level Metric 803 indicates which present characteristic of the transaction group to evaluate and the value to evaluate the characteristic against, should the condition listed under System Level Metric 801 be met. For conditions that match criterion System Level Metric 801, Transaction Grouping Method 802, and Transaction Group Level Metric 803, the Facility Action 804 is triggered. For clarity within this example, the actions to be invoked under Facility Action 804 are described in plain text. In other embodiments the Facility Action 804 may be carried on the Transaction Level Criterion Matrix 530 as an identifier referencing a logical process that would be embodied in a form executable by the facility, illustrated as Interval Criterion Actions 560 (Figure 2D).

[0043] Figure 3 is a flow diagram showing an embodiment of using an aspect of the invention beginning at step 810. At step 820, at least one criterion of a workload characteristic of the transaction processing system is defined. At step 830, at least one threshold metric for each criterion is defined (these may be system level and/or transaction level metrics). At step 840, at least one trigger action is defined to be executed in response to the at least one threshold metric being met. At step 850, at least one transaction identifier identifying subsets of transactions is defined. At step 860, the at least one transactional level threshold metric is associated with the at least one transaction identifier. In one aspect, a step may occur to check and determine when a threshold metric (or metrics) is met. At step 870, at least one trigger action is performed in response to the at least one threshold metric being met. The process then exists at step 880.

[0044] Using the invention, greatly focused and much improved control of a transaction processing system is achievable. Individual transactions may be acted upon when abnormal circumstances demand action instead of performing system-wide or more global action. The invention provides for the ability to deliver substantially improved autonomic administration of transaction and system level functions via consideration of any number and degree of workload characteristics in the decision making process. Better system performance results. Further, since the services provided in accordance with the invention occur during real-time system processing, system administrators have improved

ability to more accurately control specific deficiencies in system performance and develop criterion to overcome those deficiencies.

**[0045]** Other embodiments of the invention may include or exclude particular transactions from consideration under a given interval criterion entry. Still other embodiments of the system may allow for evaluation of system-level criterion only, without regard to transaction-level evaluations. Yet in other embodiments, the invention allows for the converse, transaction-level criterion evaluation without system-level criterion evaluation. An example implementation of either of these would be the support of a 'null' evaluation for either criterion column.

**[0046]** Yet another embodiment of the invention allows for the evaluation of groups of transactions. An example of this includes evaluation of an aggregate of a transaction-level metric data gathered from a logical grouping of transactions. An example of where a reference to such grouping logic would be carried is on a column upon the interval criterion data source. This may include, for example, classes of preferred users that might have less stringent controls, or certain transactions that are exempt from certain criteria.

**[0047]** The facility may execute at set time (i.e., predefined) increments within an embodiment of the invention. One of ordinary skill in the art may choose to establish other techniques of triggering the invocation of the facility which embodies the invention.

Further, a straightforward manner of specifying and recording the desired system behaviors that form the basis of the invention's decision tree is a useful and flexible feature of the invention.

**[0048]** The action to be taken by the invention, as referenced by the interval criterion data source, may act against a given transaction, other transactions or some other aspect of the hosting transaction processing system or another transaction processing system. An example of the invention acting upon the given transaction is the purging or quiescing of the transaction. An example of the invention acting upon another transaction is the purging or quiescing of other transactions to allow a more important transaction to have preferred access to constrained resources. Examples of the invention acting upon other aspects of the transaction processing system include suspending new transactions from entering the system, disallowing transactions from a given source, quiescing the system and triggering routing of transactions to a different transaction processing system. An example of the invention action upon another transaction processing system is the triggering of another transaction processing system to cease forwarding transactions to the host transaction processing system. Terminating a process may also occur.

**[0049]** While the invention has been described in terms of embodiments, those skilled in the art will recognize that the invention can be practiced with modifications and in the spirit and scope of the appended claims.